HACKATHON OMR CHALLENGE

TRIFORCE

Protótipo de Solução para Correção Automatizada de Provas e Simulados

Equipe:
7546 - Bianca Panacho Ferreira
8745 – Pedro Henrique Campos Moreira

Rio Paranaíba
11/2023

SUMÁRIO

1 Visão Geral da Solução

- O público alvo deste protótipo é composto por educadores, instituições de ensino, e professores, que desejam melhorar a eficiência na correção de provas e simulados. Ele pode ser usado por professores, educadores, administradores escolares, cujos interesses e expectativas foram considerados durante o desenvolvimento do sistema.

- O objetivo é fornecer uma solução que simplifique e automatize o processo de correção de provas e simulados, tornando-o mais rápido, preciso e eficiente. Ele visa melhorar a qualidade do ensino, economizando o tempo dos educadores e fornecendo feedback mais rápido aos alunos.

- O projeto contribuirá para a educação, permitindo que educadores dediquem mais tempo ao ensino e ao desenvolvimento de estratégias pedagógicas, em vez de gastar horas corrigindo manualmente provas e simulados. Além disso, o protótipo inclui relatórios individuais de cada aluno, mostrando quantas alternativas o aluno acertou, errou e a pontuação total.

- Existem várias soluções para a correção de provas, mas a maioria delas não oferece uma automação completa e personalizável. Há uma demanda significativa por uma ferramenta flexível que atenda às necessidades específicas de educadores e estudantes, de forma que não haja nenhum processo manual a ser feito.

- O escopo da solução contempla as seguintes funcionalidades:
    - Escaneamento de provas físicas;
    - Leitor ótico de marcas (OMR) para fazer uma visualização especial que consegue identificar marcas feitas a caneta preta nas posições de um formulário pré-definido (gabarito do professor e cartão resposta dos alunos);
    - Correção automática com base em chaves de resposta predefinidas (gabarito do professor);
    - Geração de relatórios individuais para cada aluno;
    - Análise estatística de desempenho por questão;
    - Suporte a questões de múltipla escolha;
    - Exportação de resultados em planilhas do Excel;
    - A Webcam, a câmera do celular ou a embutida no notebook são utilizadas para capturar imagens das provas ou simulados;
    - Interface Web, onde os usuários podem realizar login, cadastrar novos usuários, executar o programa de correção automática e visualizar os resultados.

- Evoluções futuras do sistema:
    - Correção de respostas dissertativas, verdadeiras ou falsas;
    - Integração com sistemas de gerenciamento de aprendizado (LMS).

2 Requisitos

2.1 Especificações do sistema

Especificações necessárias para o funcionamento efetivo do sistema e suas funcionalidades:

- Ter acesso a uma câmera;
- Possuir ambos os gabaritos: professor e aluno;
- É obrigatório marcar todas as alternativas no gabarito do professor e nos cartões de resposta dos alunos, não há o reconhecimento de alternativas em branco;
- Cadastrá-los nesta ordem, respectivamente.

2.2 Requisitos Funcionais

F1. Autenticação de Usuário:	**Oculto ( )**
Descrição: O sistema deve permitir que os usuários façam login usando credenciais válidas para acessar as funcionalidades do sistema.	
F2. Cadastro de Usuário:	**Oculto ( )**
Descrição: O sistema deve permitir que novos usuários se cadastrem fornecendo informações como nome, e-mail e senha.	
F3. Definição do Gabarito:	**Oculto ( )**
Descrição: Os educadores devem poder definir o gabarito da prova, especificando as respostas corretas para cada pergunta.	
F4. Captura de Imagem	**Oculto ( )**
Descrição: O sistema deve acessar a webcam do notebook, webcam ou a câmera do celular para capturar imagens do gabarito.	
F5. Correção Automatizada:	**Oculto ( )**
Descrição: O sistema deve analisar as respostas dos alunos capturadas no imagem e compará-las com o gabarito, identificando respostas corretas e incorretas.	
F6. Geração de Relatório:	**Oculto ( )**
Descrição: Após a correção automatizada, o sistema deve gerar um relatório contendo a pontuação total, número de acertos e erros de cada aluno.	

2.3 Requisitos Não Funcionais

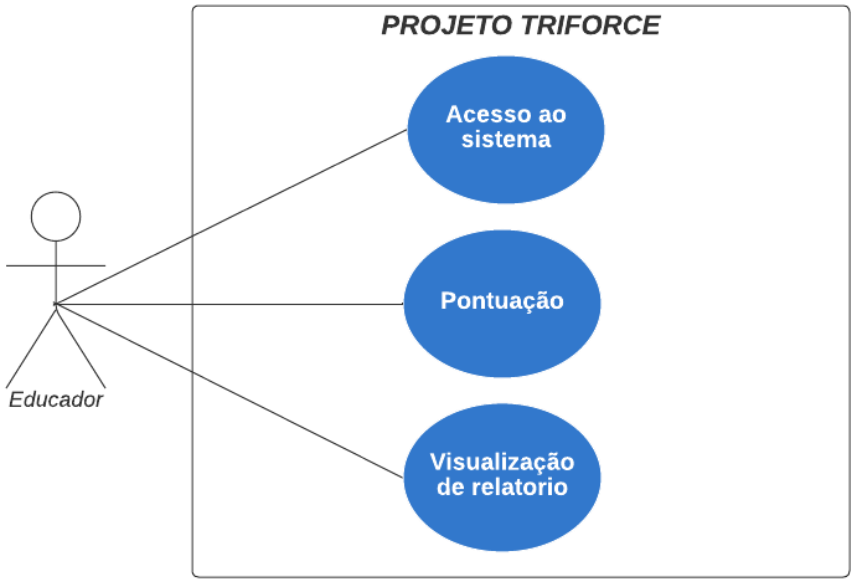
Requisitos Não-Funcionais				
Nome	**Restrição**	**Categoria**	**Desejável**	**Permanente**
Desempenho	O sistema deve ser responsivo e realizar a correção das provas em tempo real, mesmo com um grande número de alunos.	Desempenho	**( )**	**(X)**
Segurança	As informações dos usuários e dos alunos devem ser armazenadas de forma segura.	Segurança	**( )**	**(X)**
Usabilidade	A interface do usuário deve ser intuitiva e fácil de usar, permitindo que os educadores utilizem o sistema sem dificuldades.	Usabilidade	**( )**	**(X)**
Confiabilidade	O sistema deve ser estável e confiável, minimizando falhas ou interrupções durante o processo de correção das provas.	Confiabilidade	**( )**	**(X)**
Compatibilidade	O sistema deve ser compatível com diferentes dispositivos e navegadores para garantir uma experiência consistente para os usuários.	Compatibilidade	**( )**	**(X)**
Manutenibilidade	O código do sistema deve ser modular e bem documentado, facilitando futuras atualizações e manutenções.	Manutenibilidade	**( )**	**(X)**
Escalabilidade	O sistema deve ser projetado para ser escalável, permitindo que ele cresça e suporte um número crescente de usuários e provas sem comprometer o desempenho.	Confiabilidade	**( )**	**(X)**

3. Organização dos Requisitos

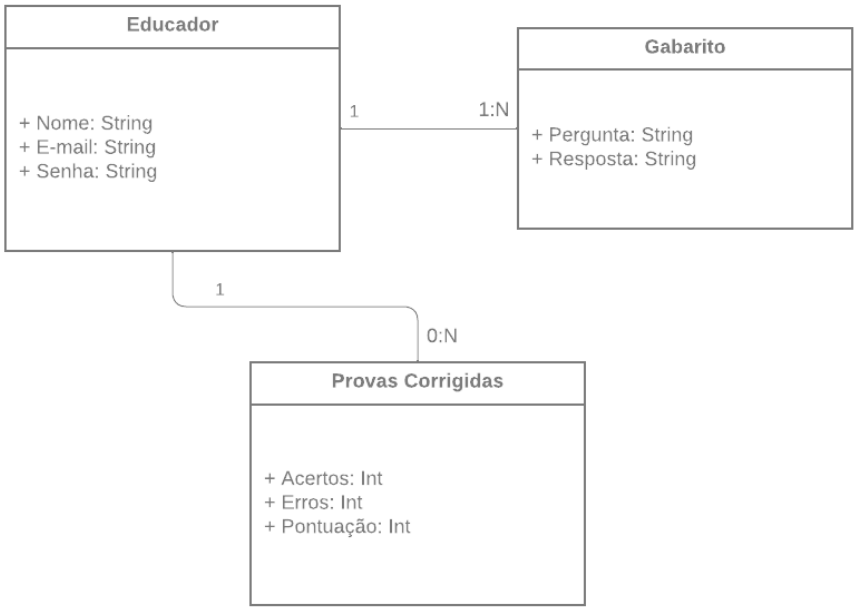
3.1 Casos de Uso

Nome	Atores	Descrição	Referências Cruzadas
Acesso ao sistema	Educador	Processo de autenticação do educador no sistema por meio de login e/ou cadastro.	F1 e F2
Pontuação	Educador	Permite que o educador cadastre o gabarito do professor e dos alunos e visualize o resultado da comparação entre ambos.	F3, F4 e F5
Visualização de Relatório	Sistema	O sistema gera um relatório de correção após a análise, contendo informações como pontuação total, número de acertos e erros de cada aluno.	F6

3.1.1 Diagrama de casos de uso



4. Modelo Conceitual do Domínio do Problema



5. Instruções de Uso para o Educador

5.1 Instruções Gabarito (Professor):

- Verifique a barra de tarefas, a janela "Gabarito" estará aberta;
- Posicione o gabarito em frente a câmera;
- Verifique se a grade das alternativas está azul;
- Pressione a tecla "s" para salvar o gabarito do professor;
- Pressione a tecla "q" para fechar a janela e abrir a do cartão-resposta.

5.2 Instruções Cartão de Respostas (Aluno):

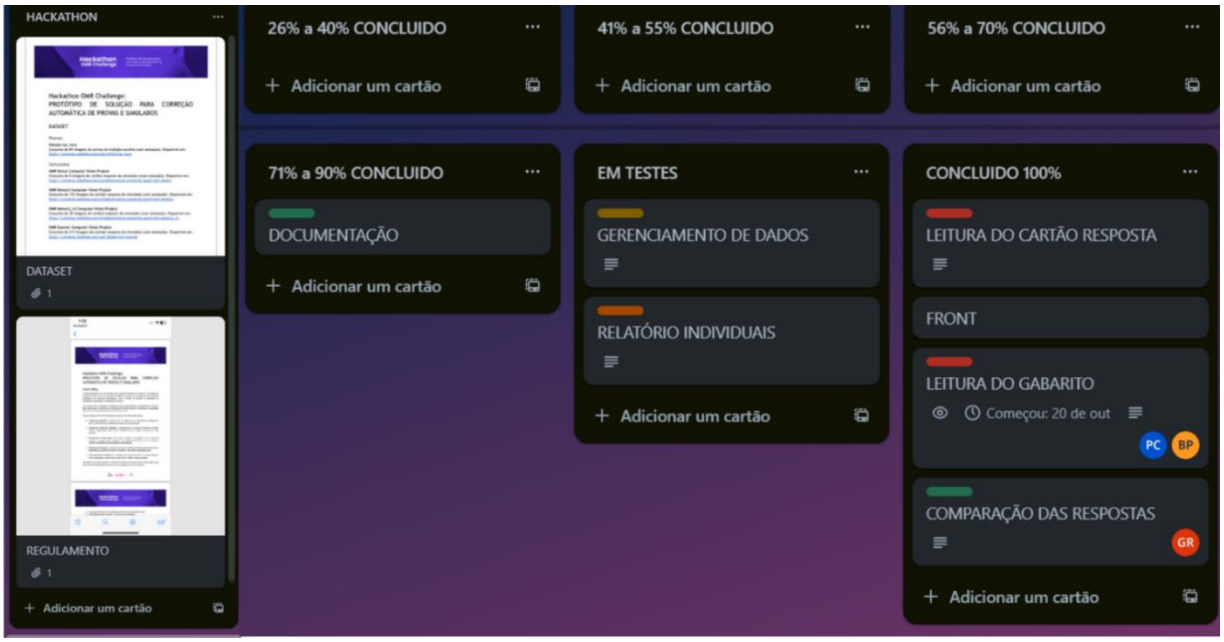
- A janela abrirá automaticamente;
- Pressione a tecla "p" para pausar e salvar o gabarito do aluno;
- Pressione "p" novamente para despausar e repetir o processo;
- Pressione a tecla "q" para fechar o sistema.

6. Metodologia Usada para Desenvolver a Solução

6.1 Scrum

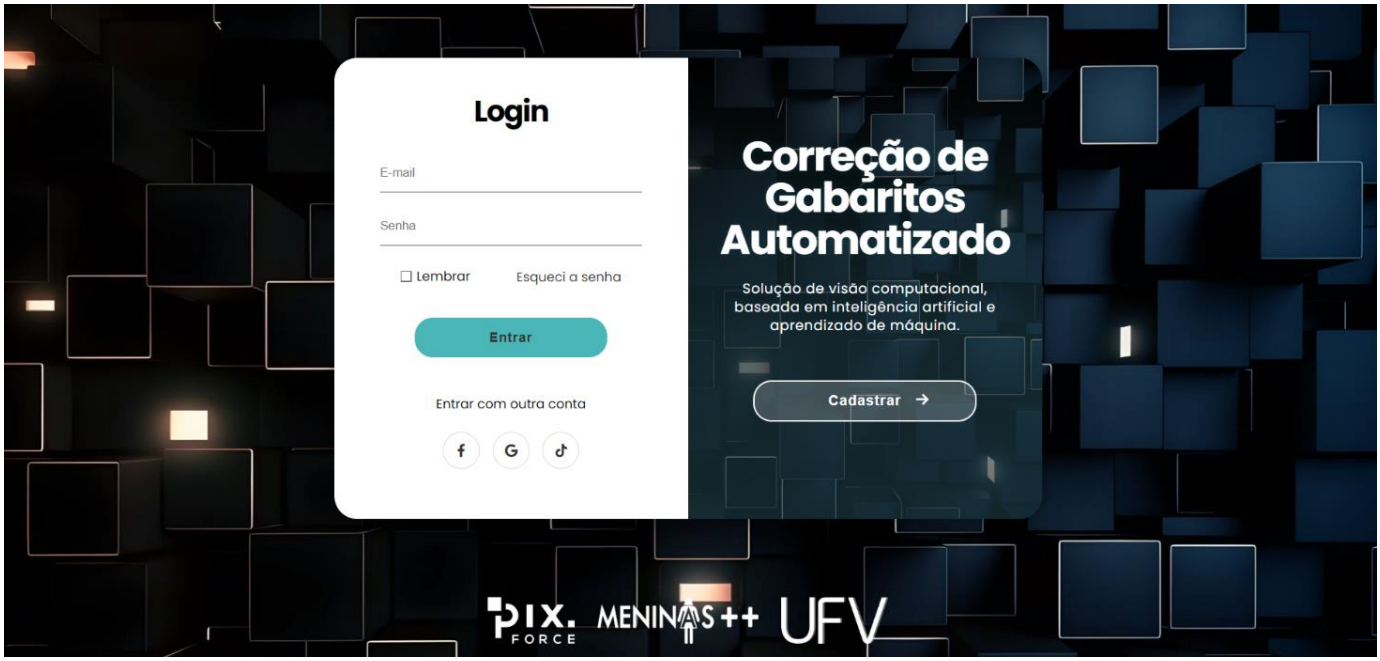
Para o desenvolvimento da solução utilizamos um processo baseado em Scrum, com reuniões diárias, semanais, e backlog para listagem dos requisitos. Utilizamos a ferramenta trello para controle do backlog:

1. **Daily:** 15 minutos diários, listagem dos backlogs feitos durante o dia e auxílio nas dúvidas, erros e tarefas incompletas.
2. **Sprint:** Semanalmente, reuniões presenciais para a atualização dos requisitos e backlogs.



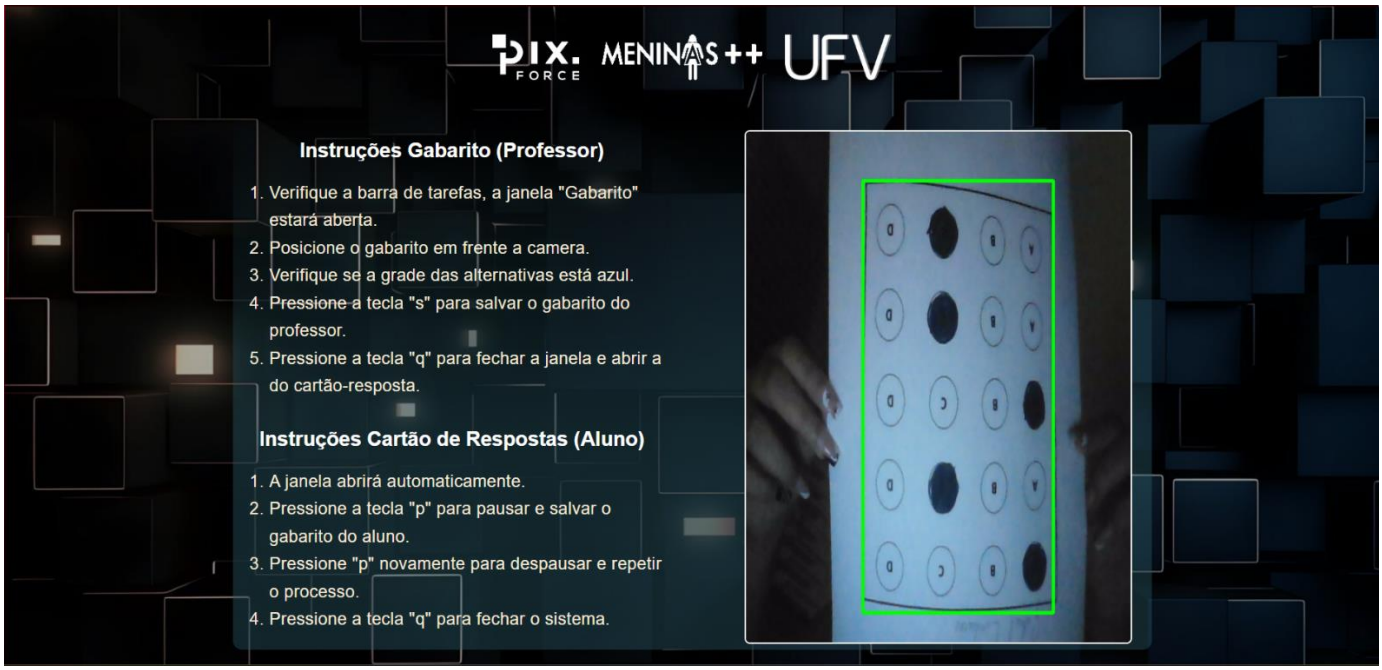
7. Projeto de Interface

7.1 Sistema web



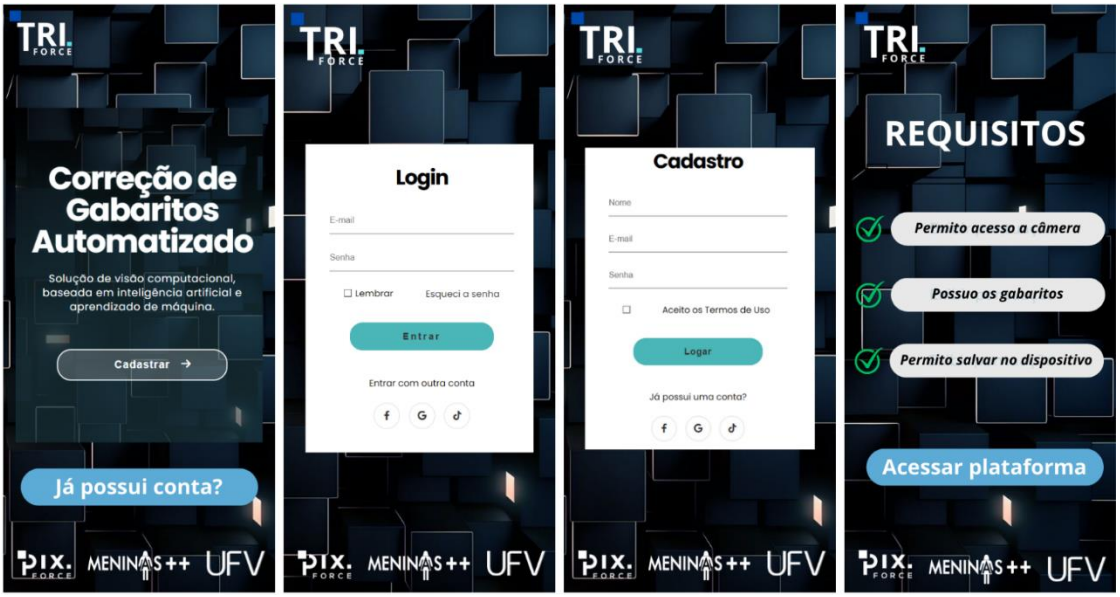
Tela de Login: Ocorre a autenticação do usuário no sistema, onde o usuário fornece credenciais (nome de usuário e senha) para acessar o sistema, e há um breve resumo a respeito das tecnologias utilizadas para a correção automatizada, como visão computacional.

Tela de Requisitos: Lista todos os requisitos necessários para que o sistema seja funcional, e há um botão de direcionamento até a plataforma onde os gabaritos serão cadastrados.



Tela de Correção: Apresenta as orientações para o usuário e auxilia no entendimento da correção, trata-se de uma janela simples e fixa no navegador, que serve para exibir a pontuação do aluno mediante o gabarito do professor cadastrado.

7.2 Sistema Mobile



8. Estruturas do código

Metodologias desenvolvidas e aplicadas ao sistema web, explicando a lógica por trás do código, exemplificando o Front-End e Back-End.

- Visão geral do projeto:

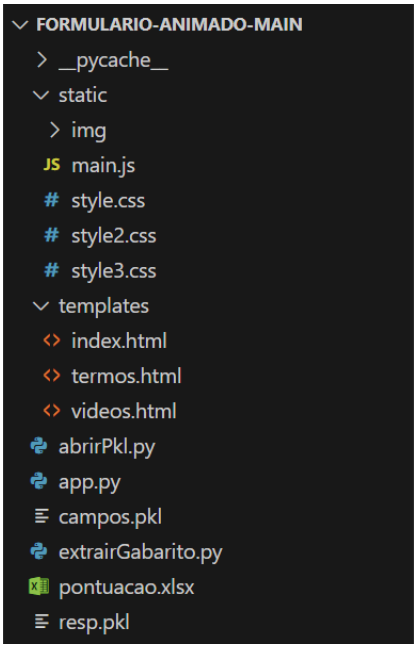
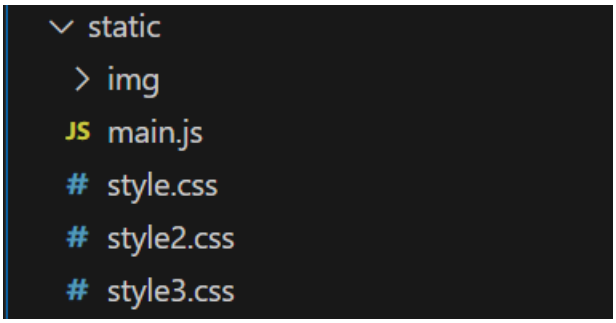
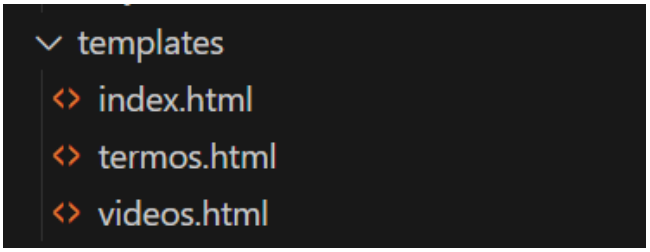


Imagem contendo todas as pastas e subpastas presentes no projeto, essa divisão foi feita baseada nas regras de implementação do **Flask**, micro framework para desenvolvimento web em Python.
Estruturação do código:

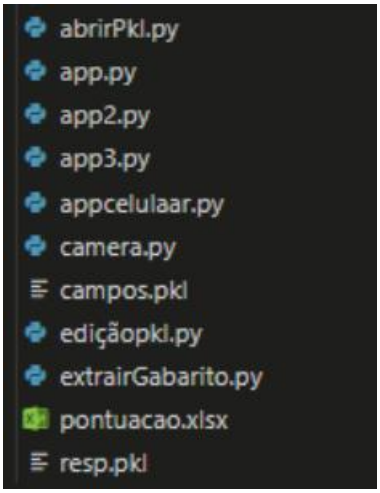
- **static** (contém todas as folhas de estilo, imagens e Javascript):
    - **img** (imagens utilizadas no projeto);
    - **main.js** (arquivo Javascript, servindo para estilização do CSS);
    - **style.css** (folha de estilo 1, linkada ao index.html);
    - **style2.css** (folha de estilo 2, linkada ao termos.html);
    - **style3.css** (folha de estilo 3, linkada ao videos.html).



- **Templates** (contém todas as páginas de Linguagem de Marcação de Hipertexto):
    - **index.html** (página de login, cadastro e aceitação dos termos de uso);
    - **termos.html** (página de requisitos e botão de acesso a plataforma);
    - **vídeos.html** (página de instruções de uso e janela que mostra a pontuação e acertos do aluno).



- **APP.PY**
    - **app.py** (arquivo Python que contém todo o código de OMR do projeto, comparação entre gabaritos e função de salvar no Excel).
    - **extrairGabarito.py** (responsável pela captura de imagens e envio ao arquivo campos.pkl).
    - **campos.pkl** (arquivo responsável pela mapeação do modelo de gabarito).
    - **resp.pkl** (arquivo responsável pela mapeação das respostas e auxílio na previsão).



- **Importação das bibliotecas**

```
import cv2
import pickle
import extrairGabarito as exG
import keyboard
import openpyxl
from flask import Flask, render_template, Response
```

OpenCV para manipulação de imagens e vídeos, é a biblioteca mais importante do código.

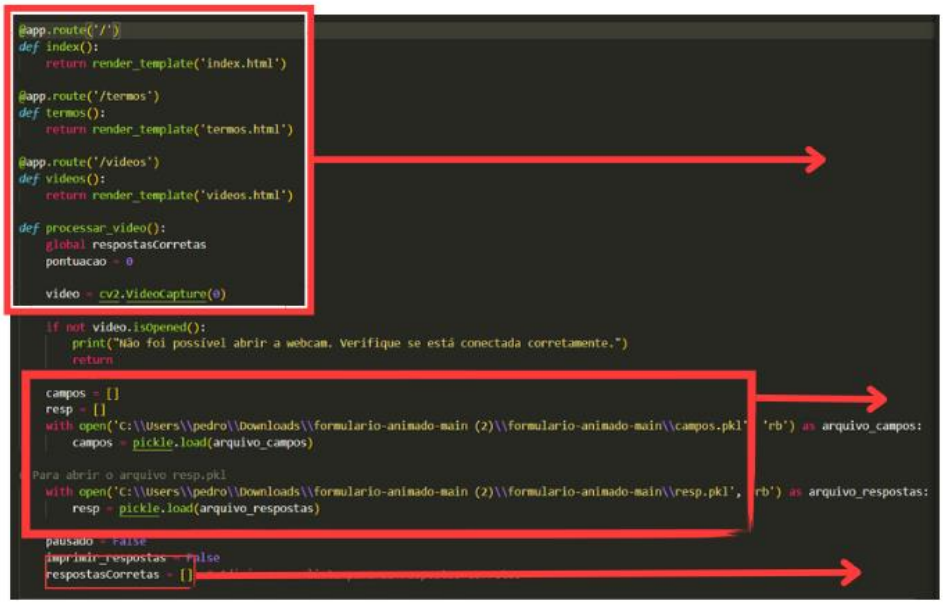
- **FLASK e variáveis globais**

```
app = Flask(__name__)

# PROFESSOR
# CADASTRAR GABARITO
camera = cv2.VideoCapture(0)

respostasCorretas = None
```

Inicialização do Flask

Inicialização da câmera e da variável de respostas corretas.

- **Definição rotas de aplicação e armazenamento das respostas**

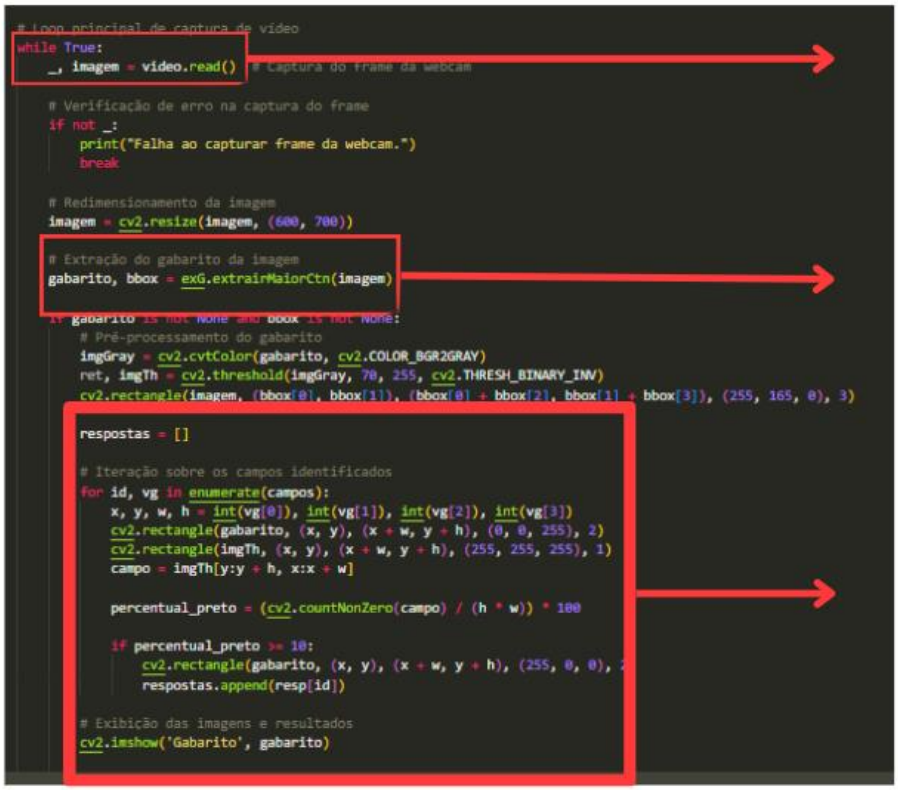


Definição das rotas de aplicação web com flask e função de encapsulamento para o 1 loop.

Leitura dos campos e respostas de arquivos pickle.

Lista para armazenar respostas corretas.

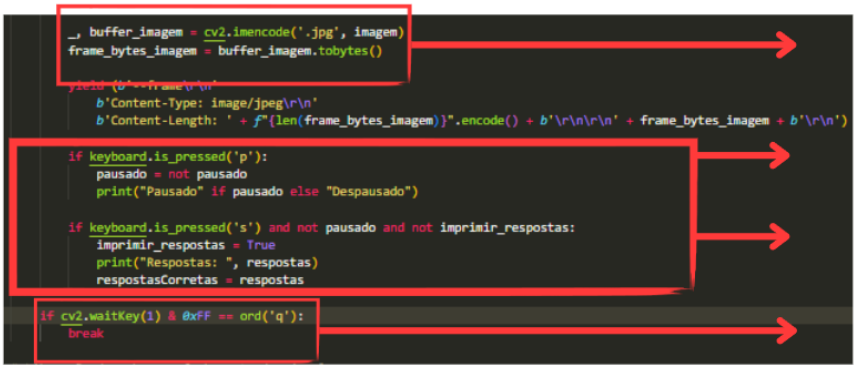
- **Loop principal de captura de vídeo**



Início do loop.

Extração do gabarito da imagem apresentada.

Interação sobre os campos que foram identificados e Exibição da imagem.



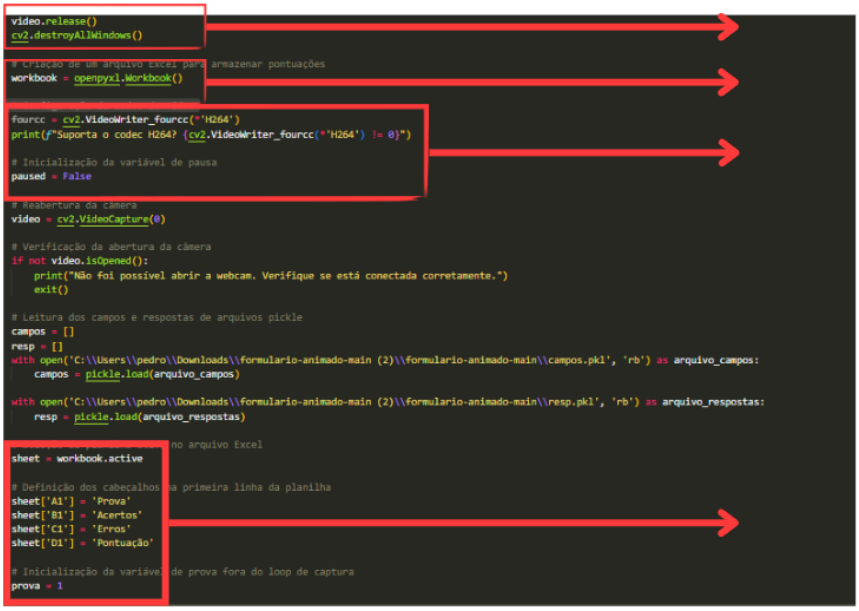
Conversão da imagem para formato JPEG e envio para a interface web.

Verificação da tecla 'p' para pausar/despausar.

Verificação da tecla 's' para imprimir respostas.

Verificação da tecla 'p' para pausar/despausar.

- **Fim do loop principal e requisitos do segundo**



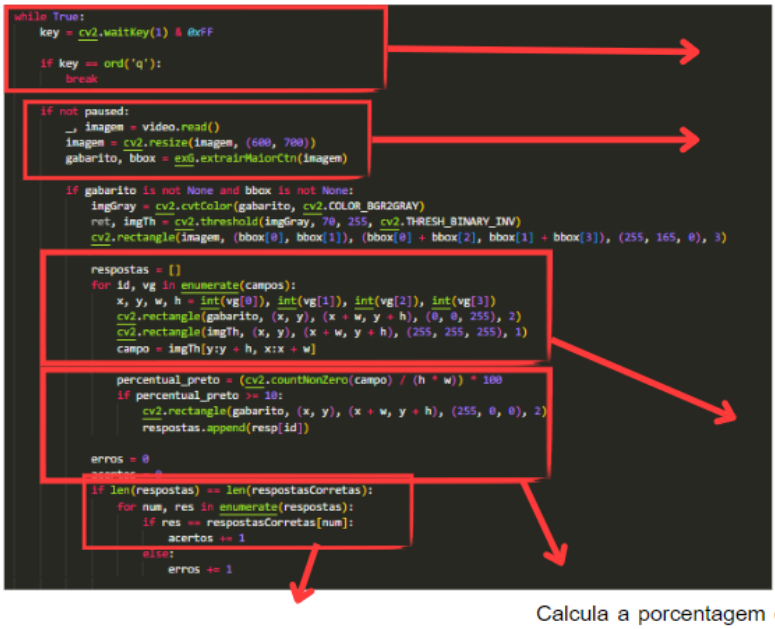
Finalização do primeiro loop.

Criação do Excel.

Configuração do codec de vídeo e inicialização da variável de pausa.

Definição do cabeçalho do Excel, seleção da planilha ativa no Excel e inicialização da variável prova.

- **Loop 2**



Inicialização do segundo loop e definição da tecla q para fechar as janelas do sistema.

Verifica se o vídeo não está pausado, o processamento ocorre apenas se o vídeo não estiver pausado.
Captura um frame do vídeo usando o método read() do objeto vídeo.
O resultado é uma tupla, o 1° elemento é um indicador de sucesso (True: leitura bem-sucedida) e o 2° elemento é a imagem.

A função está armazenando as respostas do aluno, a partir dai realiza o procedimento de correção.

Verifica se o número de respostas extraídas é igual ao número de respostas corretas.

Calcula a porcentagem de pixels pretos dentro de cada campo na imagem binária (imgTh) e, se for maior ou igual a 10%, adiciona a resposta correspondente à lista respostas: Erros = 0 e Acertos = 0. Inicializa contadores para o número de erros e acertos.

- **Finalização do código**

Definição das cores das letras que aparecem na tela do educador.

Envio da imagem para a interface web.

Atualização da planilha Excel após pressionar qualquer tecla (exceto 'q')

Salvamento do arquivo Excel, liberação da webcam e fechamento das janelas (fim do segundo loop).

Rota para o feed de vídeo e execução da aplicação Flask (final do código e do flask).

- **GADGETS**

```
cv2.VideoCapture(2)
```

```
cv2.VideoCapture('http://192.168.0.119:4747/video')
```

Estamos demostrando dois tipos de aplicações diferentes do sistema.
A primeira aplicação realiza a abertura da câmera através de uma câmera externa (WEBCAM) com variável sendo a igual dois, trocando essa variável por zero a câmera embutida do notebook é aberta.
A segunda trata-se de um link fornecido pelo aplicativo DroidCam, que o educador deve instalar em seu aparelho para utilizar a câmera do celular como forma de correção.
As 3 versões estão funcionais e disponíveis para escolha do educador antes de executar o programa.

- **Quantidade de questões e alternativas**

A parte grifada mostra como é possível atualizar a quantidade de questões e alternativas de cada prova e simulado, atendendo a solicitação do usuário.

É necessário adicionar as coordenadas das novas questões a grade ou a manta (janela Gabarito) que está no código, assim é possível corrigir qualquer gabarito desde que seja especificada a quantidade de questões e alternativas.

Os arquivos .pkl (pickle) são arquivos binários que realizam o mapeamento, criação e identificação das alternativas e questões. Para realizar modificações e implementações deve-se calcular as novas coordenadas a respeito das novas questões e alternativas, e posicioná-las entre os () apresentados no código acima.

9. Diagrama de Estados de Navegação

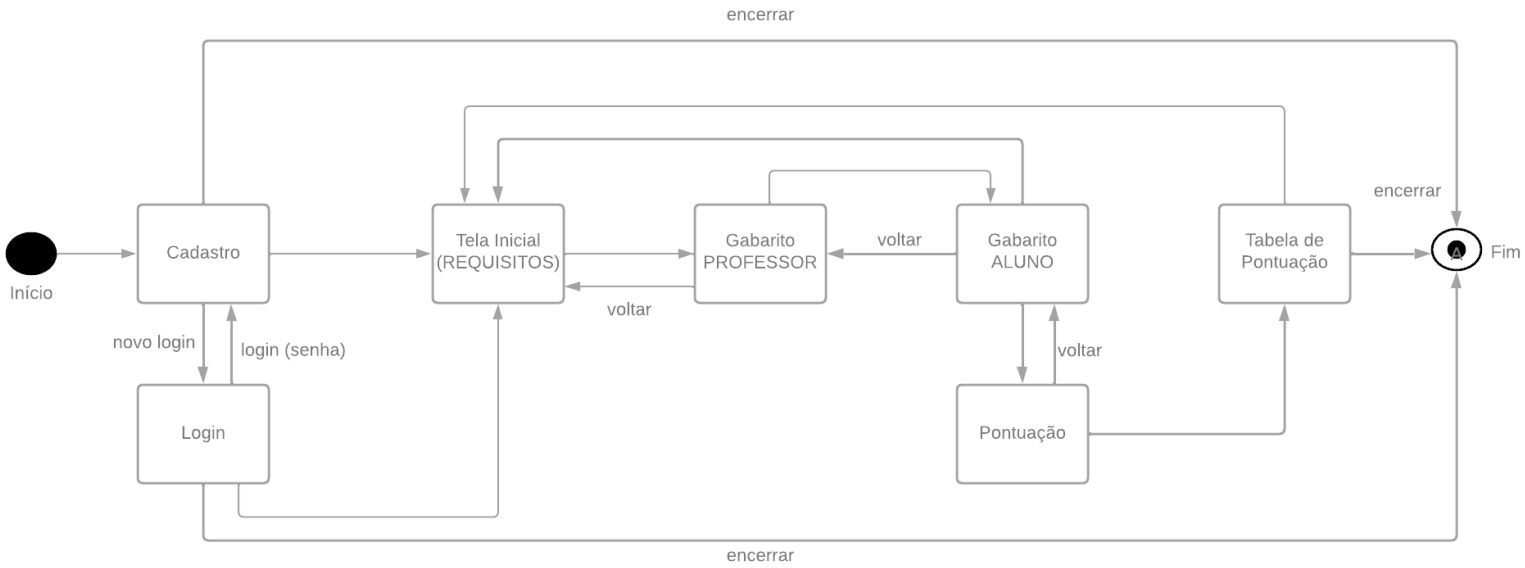


Diagrama usual para modelos web e mobile.